La coédition langue↔UNL pour partager la révision entre les langues d'un document multilingue : un concept unificateur

Christian BOITET, TSAI Wang-Ju

GETA, CLIPS, IMAG
385 rue de la Bibliothèque, BP 53
38041 Grenoble cedex 9, France
Christian.Boitet@imag.fr, Wang-Ju.Tsai@imag.fr

Résumé – Abstract

La coédition d'un texte en langue naturelle et de sa représentation dans une forme interlingue semble le moyen le meilleur et le plus simple de partager la révision du texte vers plusieurs langues. Pour diverses raisons, les graphes UNL sont les meilleurs candidats dans ce contexte. Nous développons un prototype où, dans le scénario avec partage le plus simple, des utilisateurs "naïfs" interagissent directement avec le texte dans leur langue (L0), et indirectement avec le graphe associé pour corriger les erreurs. Le graphe modifié est ensuite envoyé au déconvertisseur UNL-L0 et le résultat est affiché. S'il est satisfaisant, les erreurs étaient probablement dues au graphe et non au déconvertisseur, et le graphe est envoyé aux déconvertisseurs vers d'autres langues. Les versions dans certaines autres langues connues de l'utilisateur peuvent être affichées, de sorte que le partage de l'amélioration soit visible et encourageant. Comme les nouvelles versions sont ajoutées dans le document multilingue original avec des balises et des attributs appropriés, rien n'est jamais perdu, et le travail coopératif sur un même document est rendu possible. Du côté interne, des liaisons sont établies entre des éléments du texte et du graphe en utilisant des ressources largement disponibles comme un dictionnaire L0-anglais, ou mieux L0-UNL, un analyseur morphosyntaxique de L0, et une transformation canonique de graphe UNL à arbre. On peut établir une "meilleure" correspondance entre "l'arbre-UNL+L0" et la "structure MS-L0", une treille, en utilisant le dictionnaire et en cherchant à aligner l'arbre et une trajectoire avec aussi peu que possible de croisements de liaisons. Un but central de cette recherche est de fusionner les approches de la TA par pivot, de la TA interactive, et de la génération multilingue de texte.

Coedition of a natural language text and its representation in some interlingual form seems the best and simplest way to share text revision across languages. For various reasons, UNL graphs are the best candidates in this context. We are developing a prototype where, in the simplest sharing scenario, naive users interact directly with the text in their language (L0), and indirectly with the associated graph, to correct errors. The modified graph is then sent to the UNL-L0 deconverter and the result shown. If it is satisfactory, the errors were probably due to the graph, not to the deconverter, and the graph is sent to deconverters in other languages. Versions in some other languages known by the user may be displayed, so that improvement sharing is visible and encouraging. As new versions are added with appropriate tags and attributes in the original multilingual document, nothing is ever lost, and cooperative working on a document is rendered feasible. On the internal side, liaisons are established between elements of the text and the graph by using broadly available resources such as a L0-English or better a L0-UNL dictionary, a morphosyntactic parser of L0, and a canonical graph2tree transformation. Establishing a "best" correspondence between the "UNL-tree+L0" and the "MS-L0 structure", a lattice, may be done using the dictionary and trying to align the tree and the selected trajectory with as few crossing liaisons as possible. A central goal of this research is to merge approaches from pivot MT, interactive MT, and multilingual text authoring.

Mots-clés — Keywords

Partage de révision, représentation interlingue, coédition de texte et de graphe UNL, communication multilingue
Revision sharing, interlingual representation, coedition of text & UNL graph, multilingual communication

Introduction

Il est de plus en plus nécessaire de créer et de maintenir des documents multilingues. Dans la pratique courante, un document multilingue consiste en plusieurs fichiers monolingues parallèles, qui peuvent être de la documentation technique aussi bien que des fichiers d'aide, des fichiers de messages, ou simplement de l'information thématique mise sur la Toile et destinée à une audience multilingue (médecine, cuisine, voyage…).

La tâche est difficile, même pour un document géré de façon centralisée. D'habitude, on le crée d'abord dans une unique langue source, puis on le traduit dans plusieurs langues cibles (au moins une trentaine dans le cas de firmes comme IBM, HP ou Microsoft). Il faut avoir un moyen de garder trace des modifications, qui peuvent parfois être faites en différents lieux sur des versions en différentes langues. De temps en temps, quelqu'un doit décider des modifications à intégrer dans la nouvelle version du document. Il faut pour cela traduire en langue source les modifications faites dans d'autres langues. La nouvelle et l'ancienne version en langue source sont alors comparées en utilisant des techniques de coincidence floue, de façon à ne faire traduire que de nouveaux segments.

Le problème est encore plus aigu si les documents ne sont pas gérés de façon centralisée, car les fichiers monolingues sont alors souvent dans des formats différents (Word, EgWord, Interleaf, FileMaker, formats de divers SGBD…). A. Assimi (Al Assimi 2000, Al Assimi & Boitet 2001) a montré comment "réaligner" des documents parallèles décentralisés et leur appliquer ensuite la méthodologie esquissée plus haut.

Cependant, dans les deux cas, des traducteurs humains doivent retraduire les segments nouveaux ou modifiés, ou les réviser s'ils sont traduits par un système de TA de qualité. Contrairement à ce qu'on dit souvent, la TA de qualité existe, mais seulement dans des contextes spécifiques. Voir par exemple (Vasconcellos & León 1988).

Ce que nous aimerions, c'est faire en sorte que le travail de révision puisse être partagé entre les langues, quels que soient le domaine et le contexte.

Il est clairement impossible de refléter les changements sur un fichier en langue L0 dans les fichiers en langues L1,… Ln automatiquement et fidèlement, sans une structure intermédiaire pour faire le pont, car il faudrait au moins un aligneur parfait à granularité très fine dans le cas simple d'un changment d'article ou de nom (et encore, en supposant que le genre et le nombre restent les mêmes dans chaque version Li). Dans le cas du remplacement d'un verbe par un autre verbe ayant un régime différent dans une langue cible Li, il faudrait réanalyser la phrase en Li, la transformer en conséquence, et la regénérer sans introduire de nouvelle erreur ou imprécision, tout en gardant les améliorations manuelles éventuellement apportées lors de révisions précédentes. Ou bien, il faudrait disposer d'un système de TA plus que parfait, à savoir capable d'analyser l'énoncé modifié en L0, de le transférer, et de générerer un énoncé aussi proche que possible de l'énoncé précédent en Li, toujours en supposant que celui-ci pourrait avoir été amélioré manuellement lors d'une étape précédente.

L'approche la meilleure et la plus simple nous semble être d'utiliser un interlingua formel IL et :

- de répercuter les modifications de L0 vers l'IL,
- de regénérer vers L1,… Ln depuis l'IL.

Il faudra cependant permettre des améliorations manuelles, car la forme interlingue ne sera pas toujours présente, ou pas assez améliorable par défaut d'expressivité, et les générateurs ne seront jamais parfaits.

Nous choisissons UNL (Blanc 2001, Boguslavsky, et al. 2000, Sérasset & Boitet 1999, 2000) comme interlingua pour différentes raisons :

(1) il est spécialement conçu pour le traitement linguistique et sémantique par ordinateur,

(2) il a été dérivé avec beaucoup d'améliorations du langage pivot de H. Uchida utilisé dans ATLAS-II de Fujitsu (Uchida 1989), toujours évalué comme le système de TA anglais-japonais de meilleure qualité, avec une très grande couverture (586.000 entrées par langue),

(3) les participants du projet UNL[1] ont construit des "déconvertisseurs" d'UNL vers environ 12 langues, parmi lesquels au moins ceux allant vers l'arabe, l'indonésien, l'italien, le français, le russe, l'espagnol et le thaï sont librement accessibles pour l'expérimentation au printemps 2002,

(4) bien qu'ils soient de nature formelle, les graphes UNL (voir ci-dessous) sont assez simples à comprendre avec peu de formation et peuvent être présentés de façon localisée à des utilisateurs "naïfs" en traduisant les symboles (relations sémantiques, attributs) et les lexèmes du langage UNL par des symboles et des lexèmes de leur langue,

(5) le projet UNL a défini un format "UNL-html" intégré à html pour des fichiers contenant un document multilingue complet aligné au niveau des énoncés, et a produit un "visualiseur" qui transforme un fichier dans ce format en autant de fichiers html que de langues, et les envoie à n'importe quel navigateur web.

La représentation UNL d'un texte en langue naturelle quelconque est une liste de "graphes sémantiques" où chaque graphe exprime le sens d'un énoncé. Les nœuds contiennent chacun une unité lexicale et des attributs, et les arcs portent chacun une relation sémantique. Un sous-graphe connexe par arcs peut être distingué comme "portée" ("scope"), de sorte qu'un graphe UNL peut être en fait un hypergraphe.

Les unités lexicales d'UNL (UW[2]) représentent des (ensembles de) sens de mots, quelque chose de moins ambitieux que des concepts. Leurs dénotations sont construites de façon à être comprises intuitivement par des développeurs connaissant l'anglais, c'est à dire par tous les développeurs en TALN : une UW est un terme anglais ou un symbole spécial (nombre…) la plupart du temps complété par des restrictions sémantiques. Par exemple, l'UW "process" represente tous les sens de ce mot vu comme mot vedette (ici, verbe ou nom), et "process(icl>do, agt>person)" couvre seulement les sens de traiter, travailler sur, etc.

Les attributs sont le nombre (sémantique), le sexe, le temps sémantique[3], l'aspect, la modalité, etc., et les quelques 40 relations sémantiques sont des "cas profonds" traditionnels comme l'agent, l'objet (profond), le lieu, le but, le temps, etc.

Un façon de voir un graphe UNL correspondant à un énoncé dans la langue L est de dire qu'il représente la structure abstraite d'un énoncé anglais équivalent "vu depuis L", c'est à dire où les attributs sémantiques non nécessairement exprimés en L peuvent être absents (par exemple, l'aspect si l'on vient du français, la détermination si l'on vient du japonais, etc.).

La suite est organisée comme suit. D'abord, nous présentons des scénarios de complexité interne croissante pour la situation où quelqu'un lit un document UNL dans sa langue, le corrige, et veut transmettre que corrections aux fragments correspondants dans les autres langues. Nous étudions ensuite plus précisément la correspondance entre un texte en langue L0 et sa représentation en UNL, et montrons

---

1 http://unl.ias.unu.edu

2 Universal Word, ou Unité de Vocabulaire Virtuel

3 *Time* par opposition à *Tense*

l'avantage de la découper en 3 parties : texte ↔ treille ou "carte" morpho-syntaxique ↔ "arbre-UNL" abstrait ↔ graphe UNL. Enfin, nous présentons l'état actuel de ce travail (un site web pour l'expérimentation, une méthode pour établir la seconde partie de la correspondance), et le situons par rapport aux recherches voisines.

1 Scénarios pour partager la révision entre plusieurs langues

Supposons qu'une collection de documents multilingues est stockée sur un serveur comme une collection de fichiers multilingues en format UNL-html, ou dans toute autre forme, par exemple dans une base de données, à condition (1) qu'il soit possible de produire facilement la version dans toute langue contenue dans le document, (2) que les versions soient alignées au niveau de segments du niveau des énoncés[4], et (3) que les graphes UNL puissent être stockés et alignés avec les segments.

Voici un exemple légèrement simplifié d'un fichier en format UNL-html.

<HTML><HEAD><TITLE> Example 1 El/UNL </TITLE></HEAD><BODY> [D:dn=Mar Example 1, on= UNL French, mid=First.Author@here.com] [P] [S:1] {org:el}I ran in the park yesterday.{/org} {unl} agt(run(icl>do).@entry.@past,i(icl>person)) plc(run(icl>do).@entry.@past,park(icl>place).@def) tim(run(icl>do).@entry.@past,yesterday) {/unl} {ab}{/ab}{cn dtime=20020130-2030, deco=man} 我昨天在公園裡跑步 {/cn} {de dtime=20020130-2035, deco=man} Ich lief gestern im Park. {/de} {el}{/el}{es dtime=20020130-2031, deco=UNL-SP} Yo corri ayer en el parque.{/es} {fr dtime=20020131-0805, deco=UNL-FR}	J'ai couru dans le parc hier. {/fr} {hd}{/hd}{id}{/id}{it}{/it}{jo}{/jo}{jp}{/jp} {lv}{/lv}{mg}{/mg}{pg}{/pg}{ru}{/ru}{th}{/th} [/S][S:2] {org:el}My dog barked at me.{/org} {unl} agt(bark(icl>do).@entry.@past,dog(icl>animal)) gol(bark(icl>do).@entry.@past,i(icl>person)) pos(dog(icl>animal),i(icl>person)) {/unl}{ab}{/ab} {cn} {/cn} {de dtime=20020130-2036, deco=man} Mein Hund bellte zu mir.{/de}{el}{/el} {es}{/es} {fr dtime=20020131-0806, deco=UNL-FR} Mon chien aboya pour moi. {/fr}{hd}{/hd}{id}{/id}{it}{/it}{jo}{/jo}{jp}{/jp} {lv}{/lv}{mg}{/mg}{pg}{/pg}{ru}{/ru}{th}{/th} [/S] [/P][/D] </BODY></HTML>

Les versions françaises ont été produites par déconversion automatique, tandis que les versions allemandes et chinoises ont été traduites manuellement. Les autres sont absentes.

Le résultat du visualiseur UNL pour le français est[5] :

<HTML><HEAD><TITLE>Example 1 El/UNL</TITLE></HEAD>
<BODY>J'ai couru dans le parc hier. Mon chien aboya pour moi. </BODY></HTML>

et sera probablement affiché par un navigateur web comme :

---

[4] Comme une phrase en Li peut correspondre à 2 phrases en Lj, un segment peut contenir plus d'un énoncé.

[5] "Example" n'est pas contenu entre des balises UNL et n'est donc pas traduit.

Example 1 El/UNL

J'ai couru dans le parc hier. Mon chien aboya pour moi.

et de façon similaire pour toutes les autres langues. Dans tous les scénarios, l'utilisateur est censé lire le texte dans une vue normale, sans voir aucune balise, et vouloir à un certain moment faire une modification, par exemple déplacer "hier" après "couru" et changer "pour" en "vers". En activant un bouton ou un item de menu, il arrive alors à une interface de révision.

1.1 Révision multiple sans partage

Dans le premier scénario, nous supposons qu'il n'y a pas de graphes UNL associés aux segments. On ne peut donc pas transmettre les modifications d'une langue aux autres. Le problème consiste à transmettre les modifications pour les ajouter[6] à la forme originale du document multilingue. Mais c'est impossible en éditant les documents html présentés à l'écran, car ils n'ont pas de liens vers leur forme originale (document UNL-html ou base de données). Il faut donc une interface différente de l'interface de lecture.

Le format UNL-html est antérieur à XML, d'où les balises spéciales comme [S] et {unl}. Nous en avons dérivé un format XML équivalent, dit UNL-xml. Nous avons adapté un analyseur de UNL-html pour transformer les fichiers UNL-html en UNL-xml. L'inverse peut se faire en XSLT. En utilisant DOM, on peut alors produire différentes vues, comme celle du visualiseur UNL, une présentation bilingue ou multilingue éditable, et une interface de révision, où non seulement le texte, mais aussi le graphe UNL et éventuellement d'autres structures peuvent être manipulées directement. L'implémenation est en cours.

Prenons un exemple venant d'une expérience effectuée pour le "Forum Barcelona 2004" sur des documents en espagnol, italien, russe, français et hindi. Les parties en hindi et en russe ne sont pas montrées ici, et on a ajouté le japonais à la main. La forme XML est simplifiée.

À partir de graphes UNL corrects et complets, les déconvertisseurs actuels produisent le plus souvent des phrases correctes.

Supposons, donc pour les besoins de l'illustration, qu'un graphe UNL a été produit à partir du chinois et ne contient pas d'information définitionnelle et aspectuelle.

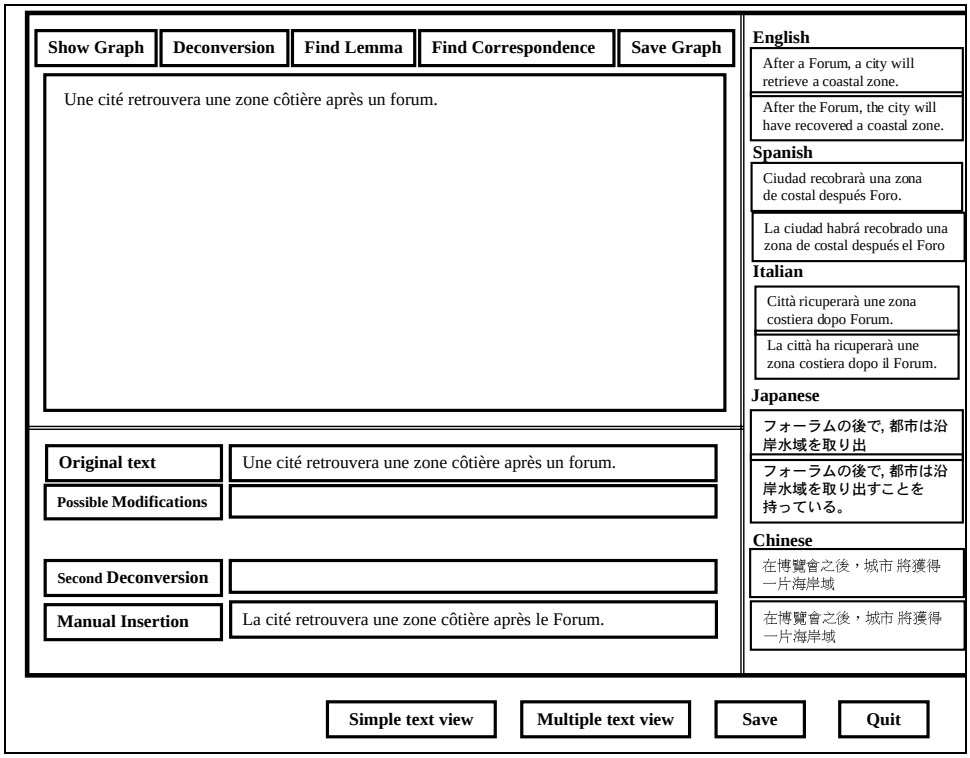
```
<unl:S num="1">
<unl:org lg="cn">在博覽會之後，城市 將獲得一片海岸域 </unl:org>
<unl:unl>
<unl:arc> agt(retrieve(icl>do).@entry.@future, city) </unl:arc>
<unl:arc> tim(retrieve(icl>do).@entry.@future, after) </unl:arc>
<unl:arc> obj(after, Forum) </unl:arc>
<unl:arc> obj(retrieve(icl>do).@entry.@future, zone(icl>place).@indef) </unl:arc>
<unl:arc> mod(zone(icl>place).@indef, coastal) </unl:arc> </unl:unl>
<unl:cn> 在博覽會之後，城市 將獲得一片海岸域 </unl:cn>
<unl:el> After a Forum, a city will retrieve a coastal zone.</unl:el>
<unl:es> Ciudad recobrará una zona de costal después Foro. </unl:es>
<unl:fr> Une cité retrouvera une zone côtière après un forum. </unl:fr>
<unl:it> Città ricuperarà une zona costiera dopo Forum. </unl:it>
<unl:jp> フォーラムの後で, 都市は沿岸水域を取り出す。 </unl:jp>
</unl:S>
```

Tous les résultats de déconversion deviennent alors faux pour les articles, et certains pour les aspects. L'interface ci-dessous, conçue pour être utilisée dans le cas du partage, peut aussi être utilisée par un lecteur connaissant plusieurs langues (affichées à la demande) et désirant pouvoir les modifier.

[6] Pour des raisons de cohérence et de sécurité, on ne peut pas envisager d'effacer les versions précédentes.

Par exemple, un hispanophone natif sachant le français et l'anglais mettrait les articles corrects ("La ciudad", "La cité", "The city", etc.) et l'aspect perfectif ("habra recobrado", "will have recovered").

Mais un francophone natif ne corrigerait probablement pas l'aspect en anglais et en espagnol, car l'aspect est souvent sous-spécifié en français, comme dans "retrouvera".



1.2 Révision transparente avec partage

Dans le second scénario, il y a un graphe UNL associé au segment modifié. Pour partager les révisions entre les langues, il faut les refléter sur le graphe UNL, i.e.

- ajouter ".@def" sur les nœuds contenant "city", "Forum".
- replacer "retrieve" par "recover" et ajouter ".@complete" sur le nœud le contenant.

Il n'est pas possible en principe de déduire une modification sur le graphe d'une modification sur le texte. Par example, le remplacement de "un" par "le" n'implique pas que le nom suivant soit déterminé (.@def), parce qu'il peut aussi être générique ("il aime la montagne"). L'approche suivie est alors :

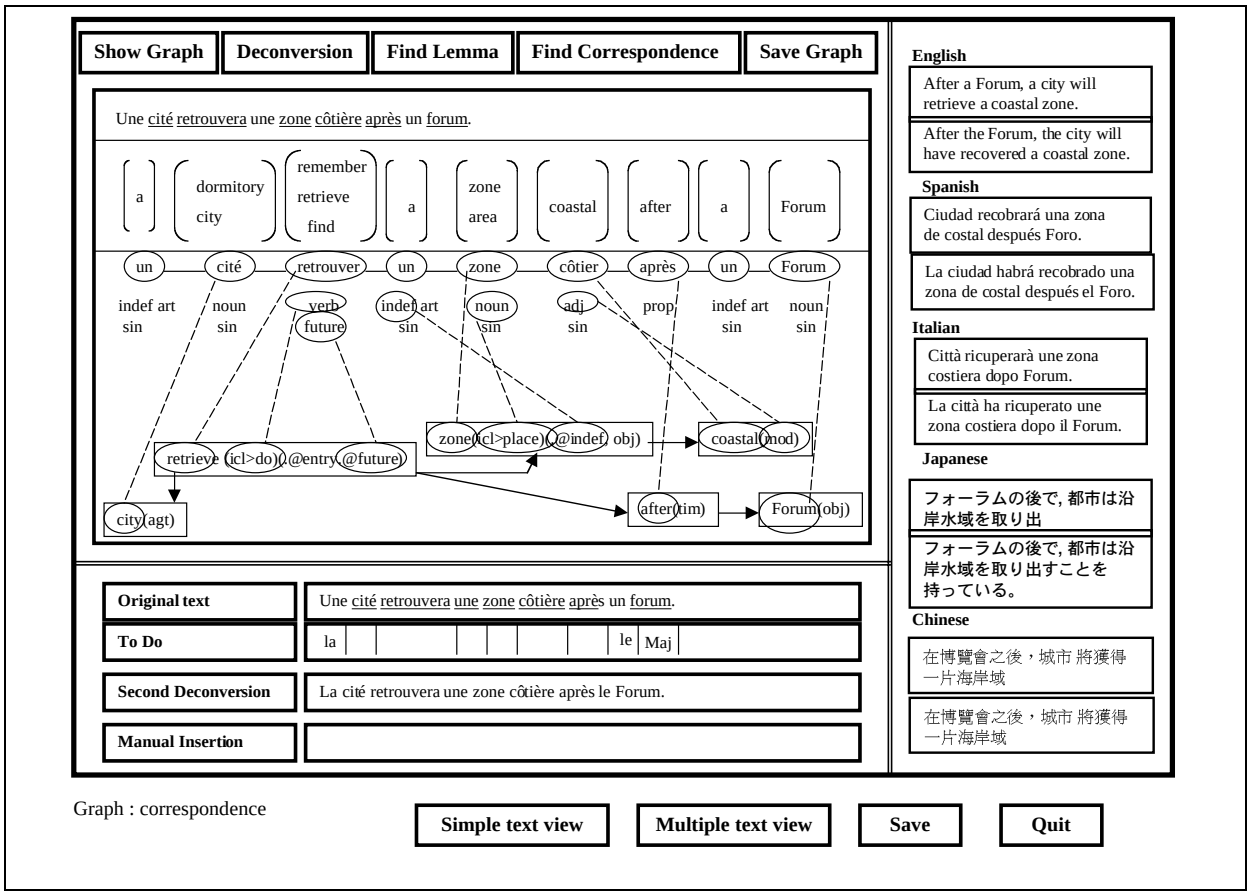
- de ne pas faire la révision en modifiant directement le texte, mais en utilisant des menus,
- de donner aux items de menu un "côté langue" visible et un "côté UNL" caché,
- quand un item de menu est choisi, de transformer seulement le graphe, de stocker l'action à effectuer sur le texte, et de la montrer à côté de son point d'application dans la zone "To Do",
- de permettre à tout moment d'envoyer le nouveau graphe au déconvertisseur L0, et d'afficher le résultat. S'il est satisfaisant, cela montrera que les erreurs étaient dues au graphe et pas au déconvertisseur, et le graphe pourra être envoyé aux déconvertisseurs d'autres langues. Les versions dans des langues Li connues par l'utilisateur pourront être affichées, de façon que l'amélioration soit visible et encourageante.

Les nouvelles versions seront ajoutées, avec des balises et des attributs appropriés, au document multilingue original en format UNL-xml, ou dans un SGBD, de façon que rien ne soit jamais perdu, et que le travail coopératif sur un document soit possible.

1.3 Révision sur plus que du texte

Pour que la méthode ci-dessus fonctionne, il faut que le texe ait été prétraité, au moins en calculant les classes morphosyntaxiques (parties du discours et attributs d'actualisation) de façon à éviter beaucoup de menus parasites, en segmentant, et en lemmatisant. Comme nous désirons que notre technique soit largement applicable, ce prétraitement doit être limité, pour pouvoir être effectué par des outils à large couverture disponibles librement pour beaucoup de langues. C'est le cas pour les analyseurs morphosyntaxiques (AMS), mais pas encore pour les analyseurs syntaxiques ou sémantiques, complets ou même superficiels *(shallow)*.

Nous proposons aussi que l'interface de révision permette d'accéder non seulement aux textes en différentes langues, mais aussi à une ou des représentations éditables du graphe UNL, du résultat de l'AMS, et de toute autre structure disponible telle qu'un arbre dérivé du graphe UNL.



Pour les utilisateurs ne souhaitant voir que du texte, le scénario précédent sera toujours utilisable. Mais il y a de bonnes raisons d'ouvrir la "boîte noire" :

1) le groupe UNL espagnol a expérimenté avec succès une interface pour la création interactive de graphes UNL, en utilisant un AMS et un éditeur de graphes montrant le graphe UNL sous une forme "localisée" (les symboles et les lexèmes UNL apparaissent en espagnol),

2) il est parfois beaucoup plus rapide de changer quelque chose dans une représentation graphique : par exemple, fusionner deux nœuds pour changer "Marie aime la fille de Marie" en "Marie aime sa fille",

3) cela peut même être nécessaire, si la correspondance est défectueuse et ne peut pas être améliorée parce que le texte est très éloigné d'une déconversion raisonnablement obtenable à partir du graphe,

4) la technologie des interfaces utilisateur a fait beaucoup de progrès, et il existe des outils de construction d'environnements conviviaux de manipulation directe,

5) enfin, les jeunes manipulent des interfaces complexes de façon très naturelle et experte, bien mieux que leurs aînés !

1.4 Ce qu'on peut et ne peut pas faire

Nous identifions quatre types d'erreurs fréquents dans le corpus que nous avons analysés :

1) des graphes contiennent de l'information fausse : mauvais attachement, mauvais choix d'UW, mauvais attribut, mauvaise relation sémantique…

2) de l'information manque dans des graphes, comme illustré ci-dessus,

3) il n'y a pas de texte parce que le graphe UNL est formellement incorrect (cela peut être dû à une manipulation humaine erronée, à une bogue dans le déconvertisseur…) : parenthèse manquante, absence de nœud d'entrée dans un graphe ou un sous-graphe…,

4) il y a des erreurs de déconversion.

Notre méthode peut être utilisée pour corriger les 2 premiers types d'erreur seulement. Si un graphe est formellement incorrect, il peut être affichable ou non. Dans le premier cas, il devrait être possible de le manipuler et de le corriger graphiquement, par exemple en connectant deux parties non reliées ou en choisissant un nœud d'entrée. Dans le second cas, il est nécessaire de travailler sur une représentation textuelle, mais c'est pour les experts (hackers) !

Si le déconvertisseur vers Li est faux, changer le graphe ne servira sans doute à rien.. Dans ce cas, l'utilisateur peut encore corriger le texte à la main (dernière zone). Ces cas devraient être enregistrés automatiquement ("logués") et envoyés plus tard comme retour (feed-back) aux développeurs, avec éventuellement des commentaires.

2 Établissement d'une correspondance texte↔graphe

2.1 Nature des correspondances

La correspondance entre un texte et un graphe UNL peut être décomposée en liaisons moins complexes, qui ne sont souvent pas de simples liens, même entre mots et nœuds. Nous avons trouvé les types suivants.

Niveau MS	graphe UNL
lemme arbre (français)	tête UW (headword) "tree"
lemme жениться (russe)	UW complète marry(agt>male)
morphème -tion (français, anglais) "男 " (chinois "nan2")	restriction d'UW (icl>action) (agt>male)

Niveau MS	graphe UNL
particule "了 "(Chinese)	attribut .@complete
trait MS d'actualisation plural	attribut .@pl
trait sémantique MS his	relation pos(*, he)

Il y a aussi des liaisons entre plusieurs mots non nécessairement connexes et un nœud (par exemple, un verbe avec particule séparable), ainsi qu'entre des attributs d'un nœud et plusieurs éléments textuels (temps+modalité et auxiliaires). On peut aussi établir des liaisons entre syntagmes et sous-graphes.

1.2 Division en trois sous-correspondances

Nous avons déjà commencé à diviser la correspondance en deux parties: texte ↔ structure-MS ↔ graphe-UNL. La structure MS peut toujours être intégrée dans un graphe acyclique contenant l'information sur les nœuds (treille) ou sur les arcs (carte ou *chart)*, de telle sorte que la première partie de la correpsondance est consituée de liaisons entre sous-chaînes (non nécessairement connexes) du texte et éléments (nœuds ou arcs) du chemin correspondant à l'interprétation préférée (en cas d'ambiguïté).

Il est peut-être possible de calculer une correspondence directe entre la structure MS et le graphe UNL, mais on ne voit pas clairement comment représenter les liaisons entre syntagmes et sous-graphes. Pour cela, une structure d'arbre est bien meilleure. Comme on ne dispose pas (encore) d'analyseur syntaxo-sémantique libre et à large couverture, pour la grande majorité des langues, on ne peut même pas utiliser un arbre produit par un analyseur superficiel. Mais il est possible d'associer un "arbre-UNL standard" à tout graphe UNL par une transformation algorithmique réversible (Blanc 2001, Boguslavsky, et al. 2000, Sérasset & Boitet 1999) : partant du nœud d'entrée extérieur, parcourir le graphe et ses sous-graphes *(scopes)* récursivement, tout en créant des nœuds auxiliaires pour les sous-graphes, des relations sémantique "inverses" pour les arcs allant dans la "mauvaise" direction, et des symboles de coindexation pour représenter la réentrance sans duplication.

On peut aussi tirer parti du fait qu'on a une structure de plus en l'enrichissant avec des unités lexicales de L0. La correspondance est maintenant divisée en 3 parties :

- texte-L0 ↔ MS-L0 (une treille ou une carte),
- MS-L0 ↔ arbre-UNL+L0 (un arbre abstrait non ordonné proche d'un arbre de dépendance), et
- arbre-UNL+L0 ↔ graphe-UNL (les liaisons peuvent être produites en modifiant la transformation réversible standard graphe→arbre).

Un autre avantage de l'introduction de cette structure d'arbre est que les correspondances entre les chaînes et les arbres ont été beaucoup étudiées (Boitet & Zaharin 1988, Vauquois & Chappuy 1985, Zaharin 1986). Elles peuvent être encodées par deux attributs exprimant ce qu'un nœud couvre lexicalement (SNODE) et syntagmatiquement, en tant que racine d'un sous-arbre (STREE).

3 État d'avancement et recherches voisines

3.1 Plate-forme expérimentale

Nous avons implémenté un site web appelé SWIIVRE-UNL (Site Web pour l'Initiation, l'Information, la Validation, la Recherche et l'Expérimentation sur UNL (Tsai 2001)) pour servir de base expérimentale à notre recherche. Il permet pour l'instant :

- d'obtenir de l'information dynamique sur les sites de déconversion UNL disponibles,
- d'accéder à une collection de documents (spécifications, articles…) sur UNL,

- de parcourir une collection de phrases et de graphes UNL alignés dans beaucoup de langues (quoique FB2002 soit le seul sous-ensemble vraiment consistant) : chapître de la Bible, aritcles de journaux sur le football, Charte de l'ONU, documentation technique…
- d'expérimenter la déconversion multilingue d'un graphe UNL en le collant simplement dans d'essayer la première version d'un éditeur de graphes UNL orienté vers le web et XML, et programmé en utilisant plus de balises (UNL-xml-ed), DOM, et javascript (Jitkue 2001). L'interface de révision esquissé ci-dessus est en construction par trois étudiants de DESS.

3.2 Construction de la correspondance treille↔arbre

Voici la première méthode, en cours d'implémentation, que nous avons trouvée pour établir une "meilleure" correspondance.

On commence avec une treille MS-L0 reliée au texte et un arbre-UNL produit de façon standard et relié au graphe UNL. Le but est d'établir des liaisons entre la treille et l'arbre, et d'ordonner l'arbre de façon à ce qu'il soit "maximalement aligné" avec la treille, et donc avec le texte. Une liaison est simplement, comme dans un diagramme entité-relation, une "boîte de relation" avec une liste ordonnée de liens vers la treille, et une autre vers l'arbre. Supposons aussi, a minima, qu'on dispose seulement d'un dictionnaire L0-anglais, et pas de dictionnaires L0-UNL.

D'abord, on enrichit la treille avec les lemmes anglais et l'arbre avec des lemmes de L0, ce qui transforme ces deux structures en MS-L0+EN et arbre-UNL+L0.

Ensuite, on établit des liens entre les nœuds de la treille et de l'arbre ayant des lemmes en commun (en L0 ou en anglais), on calcule un score pour chaque chemin dans la treille, et on choisit le meilleur.

La phase suivante consiste à aligner l'arbre avec ce chemin, en utilisant les liens "sûrs" comme point de départ, et les contraintes sur les liaisons STREE et SNODE : s'il y a des liens qui se croisent, ce qui est possible si deux mots du texte ont des sens similaires, on donne la préférence au lien qui maximise la proximité dans l'arbre et dans le texte.

On établit ensuite les liaisons des autres types : lexèmes avec relations sémantiques (à, dans… pour *plc* ou *plt ;* de, depuis pour *org* ou *plf*…), lexèmes avec attributs UNL, et attributs MS avec attributs UNL.

3.3 Recherches voisines

L'envoi automatique de retours aux développeurs se fait déjà dans certains systèmes de TA, en particulier à Taiwan (EKS) et à la PAHO[7] (Vasconcellos & León 1988), mais devrait être beaucoup plus pratiqué.

L'idée de la coédition n'est pas nouvelle : l'UPM (Madrid) l'utilise pour créer des graphes UNL, Y. Lepage à ATR et Tand E. K. at l'USM (Penang) ont développé des éditeurs de correspondances chaîne-arbre, Watanabe à IBM-Japon a montré une très jolie interface pour éditer du texte à partir de sa structure de dépendance sous-jacente, le système MULTIMETEO (Coch & Chevreau 2001) est en fait un système de coédition pour les prévisions météo et leurs structures sémantiques sous-jacentes, en six langues, et il y a un projet à Xerox visant la génération multilingue et la normalisation de textes libres dans des domaines et des typologies restreints comme les notices pharmaceutiques.

---

[7] Pan American Health Organization (Washington, D. C. et Genève)

Dans notre cas, il y a plusieurs différences majeures : (1) la coédition doit être faite du côté du consommateur, et pas, comme à l'UPM, du côté du producteur ; (2) il n'y a pas de domaine ou de typologie spécifique ; (3) l'idée de dériver un arbre sémantique abstrait d'un texte à partir d'une représentation interlingue en utilisant des techniques d'alignement et de satisfaction de contraintes, et pas un système de règles intégré à un générateur, ni a fortiori quelque analyseur que ce soit, semble nouvelle.

Conclusion

La coédition d'un texte en langue naturelle et de sa représentation dans une certaine forme interlingue paraît être la meilleure façon de partager la révision entre plusieurs langues, ce qu'on n'a encore jamais imaginé pouvoir faire. Pour différentes raisons les graphes UNL sont les meilleurs candidats.

Nous avons décrit une approche où, dans le scénario avec partage le plus simple, des utilisateurs "naïfs" interagissent directement avec le texte dans leur langue (L0), et indirectement avec le graphe associé pour corriger les erreurs. Le graphe modifié est ensuite envoyé au déconvertisseur UNL-L0 et le résultat est affiché. S'il est satisfaisant, les erreurs étaient probablement dues au graphe et non au déconvertisseur, et le graphe est envoyé aux déconvertisseurs vers d'autres langues. Les versions dans certaines autres langues connues de l'utilisateur peuvent être affichées, de sorte que le partage de l'amélioration soit visible et encourageant. Comme les nouvelles versions sont ajoutées dans le document multilingue original avec des balises et des attributs appropriés, rien n'est jamais perdu, et le travail coopératif sur un même document est rendu possible.

Le prototype en cours d'implémentation permettra un second scénario, dans lequel on pourra aussi voir et manipuler directement le graphe UNL donné, une treille ou une "carte" produite par tout analyseur morphosyntaxique libre disponible pour L0, et un arbre abstrait produit non pas par analyse, mais par une transformation standard du graphe UNL suivie par un enrichissement lexical en L0, et un alignement avec la treille ou la carte, et donc avec le texte.

Du côté interne, des liaisons sont établies entre des éléments du texte et du graphe en utilisant des ressources largement disponibles comme un dictionnaire L0-anglais, ou mieux L0-UNL, un analyseur morphosyntaxique de L0, et une transformation canonique de graphe UNL à arbre. On peut établir une "meilleure" correspondance entre "l'arbre-UNL+L0" et la "structure MS-L0", une treille, en utilisant le dictionnaire et en cherchant à aligner l'arbre et un chemin avec aussi peu que possible de croisements de liens.

Un but central de cette recherche est de fusionner les approches de la TA par pivot, de la TA interactive, et de la génération multilingue de texte. La coédition apparaît comme le concept unificateur.

Références

Al Assimi A.-B. (2000) Gestion de l'évolution non centralisée de documents parallèles multilingues. *Nouvelle thèse*, UJF, Grenoble, 31/10/00, 200 p.

Al Assimi A.-B., & Boitet C. (2001) Management of Non-Centralized Evolution of Parallel Multilingual Documents. *Proc. of Internationalization Track, 10th International World Wide Web Conference*, Hong Kong, May 1-5, 2001, 7 p.

Blanc E. (2001) From graph to tree : Processing UNL graph using an existing MT system. *Proc. of First UNL Open Conference - Building Global Knowledge with UNL*, Suzhou, China, 18-20 Nov. 2001, UNDL (Geneva), 6 p.

Boguslavsky I., Frid N., Iomdin L., Kreidlin L., Sagalova I., & Sizov V. (2000) Creating a Universal Networking Language Module within an Advanced NLP System. *Proc. of COLING-2000*, Saarbrücken, 31/7—3/8/2000, ACL & Morgan Kaufmann, 1/2, pp. 83-89.

Boitet C., & Zaharin Y. (1988) Representation trees and string-tree correspondences. *Proc. of COLING-88, Budapest, 22–27 Aug. 1988*, ACL, pp. 59—64.

Boitet C. (1999) A research perspective on how to democratize machine translation and translation aids aiming at high quality final output. *Proc. of MT Summit VII*, Singapore, 13-17 September 1999, Asia Pacific Ass. for MT, pp. 125—133.

Boitet C. (2001) Four technical and organizational keys for handling more languages and improving quality (on demand) in MT. *Proc. of MTS2001 Workshop on "MT2010 — Towards a Road Map for MT"*, Santiago de Compostela, 18/9/01, IAMT, 8 p.

Coch J., & Chevreau K. (2001) Interactive Multilingual Generation. *Proc. of CICLing-2001 (Computational Linguistics and Intelligent Text Processing)*, Mexico, February 2001, Springer, pp. 239-250.

Jitkue P. (2001) Participation au projet SWIIVRE-UNL et première version d'un environnement Web de déconversion multilingue et d'éditeur UNL de base. *Rapport de stage de Maîtrise d'informatique*, Université Joseph Fourier, septembre 2001, 13 p.

Sérasset G., & Boitet C. (1999) UNL-French deconversion as transfer & generation from an interlingua with possible quality enhancement through offline human interaction. *Proc. of MT Summit VII*, Singapore, 13-17 September 1999, Asia Pacific Ass. for MT, pp. 220—228.

Sérasset G., & Boitet C. (2000) On UNL as the future "html of the linguistic content" & the reuse of existing NLP components in UNL-related applications with the example of a UNL-French deconverter. *Proc. of COLING-2000*, Saarbrücken, 31/7—3/8/2000, ACL & Morgan Kaufmann, 2/2, pp. 768—774.

Tsai W.-J. (2001) SWIIVRE — a web site for the Initiation, Information, Validation, Research and Experimentation on UNL (Universal Networking Language). *Proc. of First UNL Open Conference : Building Global Knowledge with UNL*, Suzhou, China, 18-20 Nov. 2001, UNDL (Geneva), 8 p.

Uchida H. (1989) ATLAS. *Proc. of MTS-II (MT Summit)*, Munich, 16-18 août 1989, pp. 152-157.

Vasconcellos M., & León M. (1988) SPANAM and ENGSPAN : Machine Translation at the Pan American Health Organization. In *Machine Translation systems*, edited by Slocum, Cambridge Univ. Press, pp. 187—236.

Vauquois B., & Chappuy S. (1985) Static grammars: a formalism for the description of linguistic models. *Proc. of TMI-85 (Conf. on theoretical and metholodogical issues in the Machine Translation of natural languages)*, Aug. 1985, pp. 298-322.

Zaharin Y. (1986) Strategies and heuristics in the analysis of a natural language in Machine Translation. *Proc. of COLING-86*, Bonn, Aug. 1986, pp. 136—139.

-o-o-o-o-o-o-o-o-o-